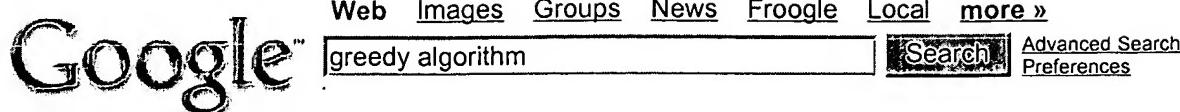


Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S55	98	coverage adj metrics	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/05 12:22
S1	2	model adj check\$3 same dominating adj block	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/05 12:22
S54	7	subset adj cover same block	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:41
S53	64	subset adj cover	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:41
S52	1	coverage same tasks same dominating	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:40
S51	2	coverage same tasks same (control\$1flow)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:38
S50	1364	coverage same tasks	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:10
S49	137430	coverage	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 14:09
S48	5	(control-flow (control adj flow)) with basic adj block same rule	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 11:51
S47	30	S45 and S46	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 11:50

S46	329	(control-flow (control adj flow)) with basic adj block	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 11:50
S45	137430	coverage	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/10/04 11:49

**Web**Results 1 - 10 of about 1,170,000 for greedy algorithm. (0.12 seconds)Ideas, Concepts and Definitions

Greedy Algorithm. An **algorithm** is a step-by-step recipe for solving a problem.

A **greedy algorithm** might also be called a "single-minded" **algorithm** or an ...

www.c3.lanl.gov/mega-math/gloss/compute/greedy.html - 4k - Oct 3, 2005 - [Cached](#) - [Similar pages](#)

greedy algorithm

Definition of **greedy algorithm**, possibly with links to more information and implementations.

www.nist.gov/dads/HTML/greedyalgo.html - 4k - [Cached](#) - [Similar pages](#)

Greedy Algorithms

"**Greedy algorithms** work in phases. In each phase, a decision is made that appears

... If the best answer is not required, then simple **greedy algorithms** are ...

www.cs.man.ac.uk/~graham/cs2022/greedy/ - 4k - [Cached](#) - [Similar pages](#)

Greedy algorithm - Wikipedia, the free encyclopedia

Greedy algorithms do not consistently find the globally optimal solution, ...

If a **greedy algorithm** can be proven to yield the global optimum for a given ...

en.wikipedia.org/wiki/Greedy_algorithm - 12k - [Cached](#) - [Similar pages](#)

Greedy Algorithms

Greedy algorithms are used to solve optimization problems. **Greedy Approach**.

Greedy Algorithm works by making the decision that seems most promising at any ...

www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Greedy/greedyIntro.htm - 9k -

[Cached](#) - [Similar pages](#)

Greedy Algorithm -- From MathWorld

... a_2, \dots, a_k with $a_1 < a_2 < \dots < a_k$, a **greedy algorithm** can be used to find

a vector of coefficients (c_1, c_2, \dots, c_k) such that ...

mathworld.wolfram.com/GreedyAlgorithm.html - 26k - [Cached](#) - [Similar pages](#)

A greedy algorithm for aligning DNA sequences.

We introduce a new **greedy alignment algorithm** with particularly good performance

and show that it computes the same alignment as does a certain dynamic ...

www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=10890397&dopt=Citation -

[Similar pages](#)

A Greedy Algorithm for Tolerating Defective Crosspoints in NanoPLA ...

Including fanin bounding, our **greedy mapping algorithm** maps a large set of benchmark designs with 13% average overhead for random junction defect rates as ...

www.cs.caltech.edu/research/ic/abstracts/xpoint_defect_fpt2004.html - 4k - [Cached](#) - [Similar pages](#)

[PDF] Microsoft PowerPoint - Greedy.ppt

File Format: PDF/Adobe Acrobat - [View as HTML](#)

The **Greedy Method**. Technique. The **greedy** method is a general **algorithm** ...

The **Greedy Method**. 7. The Fractional Knapsack. **Algorithm** ...

www3.algorithmdesign.net/handouts/Greedy.pdf - [Similar pages](#)

[PDF] Greedy Algorithms for Minimum Spanning Tree

File Format: PDF/Adobe Acrobat - [View as HTML](#)

 **PORTAL**
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login
 Search: The ACM Digital Library The Guide
 [+ "code coverage" +"model checking"] **SEARCH**

THE ACM DIGITAL LIBRARY

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [code coverage](#) [model checking](#)

Found 41 of 161,645

Sort results by Save results to a Binder
 Display results Search Tips
 Open results in a new window

Try an [Advanced Search](#)
 Try this search in [The ACM Guide](#)

Results 1 - 20 of 41

Result page: **1** [2](#) [3](#) [next](#)Relevance scale **1 Coverage estimation for symbolic model checking**

Yatin Hoskote, Timothy Kam, Pei-Hsin Ho, Xudong Zhao

June 1999 **Proceedings of the 36th ACM/IEEE conference on Design automation**Full text available:  [pdf\(79.32 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**2 Model checking Java programs using structural heuristics**

Alex Groce, Willem Visser

July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis, Volume 27 Issue 4**Full text available:  [pdf\(229.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We describe work in introducing heuristic search into the Java PathFinder model checker, which targets Java bytecode. Rather than focusing on heuristics aimed at a particular kind of error (such as deadlocks) we describe heuristics based on a modification of traditional branch coverage metrics and other structure measures, such as thread inter-dependency. We present experimental results showing the utility of these heuristics, and argue for the usefulness of *structural heuristics* as ...

Keywords: coverage metrics, heuristics, model checking, testing**3 Test input generation with java PathFinder**

Willem Visser, Corina S. Păsăreanu, Sarfraz Khurshid

July 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis, Volume 29 Issue 4**Full text available:  [pdf\(225.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We show how model checking and symbolic execution can be used to generate test inputs to achieve structural coverage of code that manipulates complex data structures. We focus on obtaining branch-coverage during unit testing of some of the core methods of the red-black tree implementation in the Java **TreeMap** library, using the Java PathFinder model checker. Three different test generation techniques will be introduced and compared, namely, straight model checking of the code, model checki ...

Keywords: coverage, model checking, red-black trees, symbolic execution, testing object-oriented programs

 **PORTAL** [Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

USPTO **Search:** The ACM Digital Library The Guide

SEARCH

THE ACM DIGITAL LIBRARY

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used [statement coverage dominator](#)

Found 5 of 161,645

Sort results by Save results to a Binder
 Display results Search Tips
 Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 5 of 5

Relevance scale **1 Structural Testing Based on Minimum Kernels**

Elena Dubrova

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 2**Full text available:  [pdf\(140.46 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Structural testing techniques, such as statement and branch coverage, play an important role in improving dependability of software systems. However, finding a set of tests which guarantees high coverage is a time-consuming task. In this paper we present a technique for structural testing based on kernel computation. A kernel satisfies the property that any set of tests which executes all vertices (edges) of the kernel executes all vertices (edges) of the program's flowgraph. We present a linear ...

2 A static measure of a subset of intra-procedural data flow testing coverage based on node coverage

Ettore M. Merlo, Giuliano Antoniol

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative research**Full text available:  [pdf\(225.32 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In the past years, a number of research works, which have been mostly based on pre and post dominator analysis, have been presented about finding subsets of nodes and edges (called "unrestricted subsets") such that their traversal during execution (if feasible) exercises respectively all feasible nodes and edges in a Control Flow Graph (CFG). This paper presents an approach to statically measure a subset of intra-procedural data flow ("all uses") coverage obtained by exercising an "unrestricted s ...

3 Efficient coverage testing using global dominator graphs

Hira Agrawal

September 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1999 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, Volume 24 Issue 5**Full text available:  [pdf\(906.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Coverage testing techniques, such as statement and decision coverage, play a significant role in improving the quality of software systems. Constructing a thorough set of tests that yield high coverage, however, is often a very tedious, time consuming task. In this paper we present a technique to find a small subset of a program's statements and decisions with the property that covering the subset implies covering the rest. We introduce the notion of a mega block which is a set of basic blocks s ...